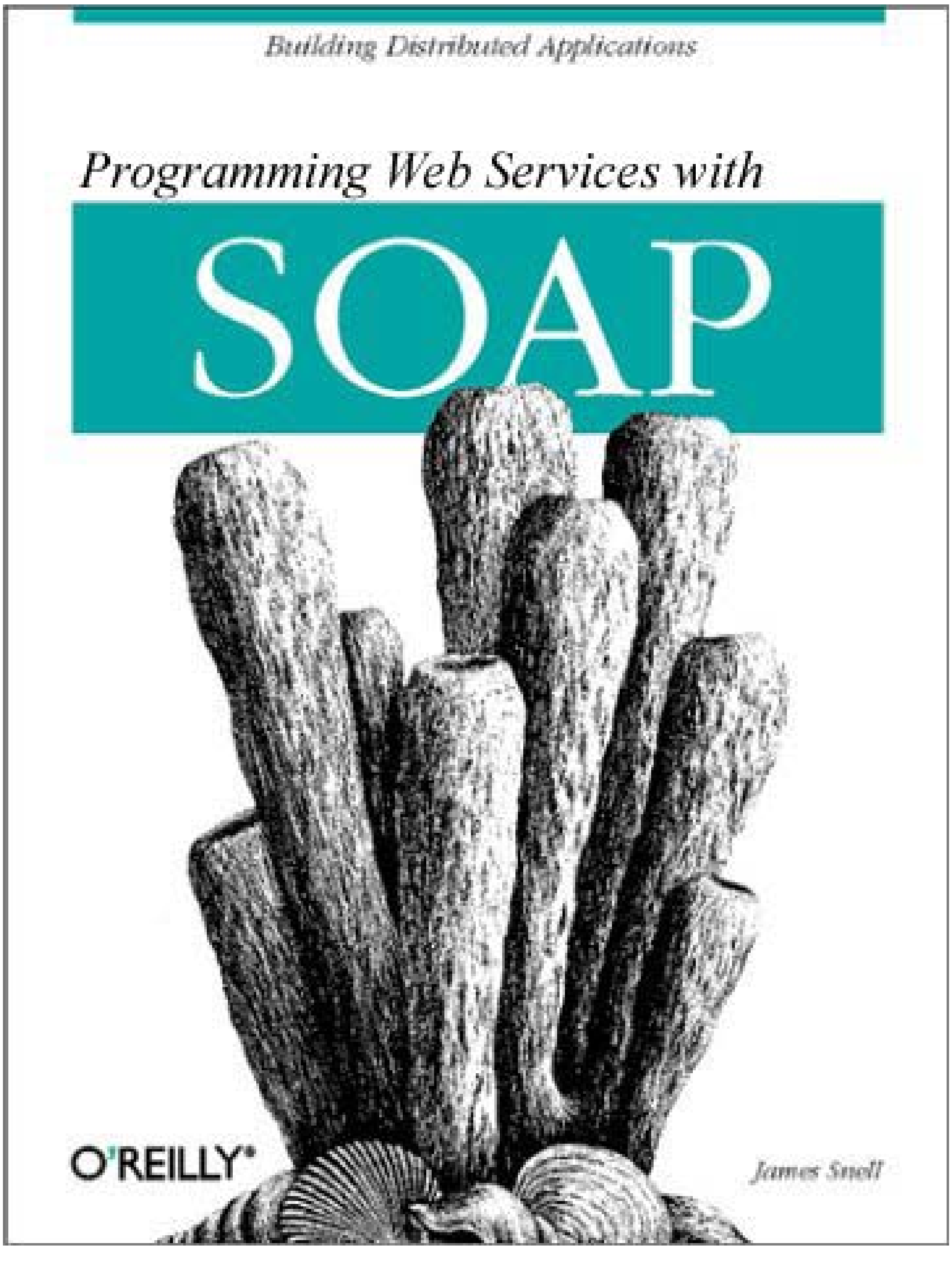

Building Distributed Applications

Programming Web Services with SOAP

O'REILLY®

James Snell

Programming Web Services with SOAP

Doug Tidwell
James Snell
Pavel Kulchenko

Publisher: O'Reilly

First Edition December 2001
ISBN: 0-596-00095-2, 216 pages

Programming Web Services with SOAP introduces you to building distributed Wb-based applications using the SOAP, WSDL, and UDI protocols. You'll learn the XML underlying these standards, as well as how to use the popular toolkits for Java and Perl. The book also addresses security and other enterprise issues.

Table of Contents

Preface

You'd be hard-pressed to find a buzzword hotter than *web services*. Breathless articles promise that web services will revolutionize business, open new markets, and change the way the world works. Proponents call web services "The Third-Generation Internet," putting them on a par with email and the browseable web. And no protocol for implementing web services has received more attention than SOAP, the Simple Object Access Protocol.

This book will give you perspective to make sense of all the hype. When you finish this book, you will come away understanding three things: what web services are, how they are written with SOAP, and how to use other technologies with SOAP to build web services for the enterprise.

While this book is primarily a technical resource for software developers, its overview of the relevant technologies, development models, standardization efforts, and architectural fundamentals can be easily grasped by a nontechnical audience wishing to gain a better understanding of this emerging set of new technologies.

For the technical audience, this book has several things to offer:

- A detailed walk-through of the SOAP, WSDL, UDDI, and related specifications
- Source code and commentary for sample web services
- Insights on how to address issues such as security and reliability in enterprise environments

Web services represent a powerful new way to build software systems from distributed components. But because many of the technologies are immature or only address parts of the problem, it's not a simple matter to build a robust and secure web service. A web service solution today will either dodge tricky issues like security, or will be developed using many different technologies. We have endeavored to lay a roadmap to guide you through the many possible technologies and give you sound advice for developing web services.

Will web services revolutionize everything? Quite possibly, but it's not likely to be as glamorous or lucrative, or happen as quickly as the hype implies. At the most basic level, web services are plumbing, and plumbing is never glamorous. The applications they make possible may be significant in the future, and we discuss Microsoft Passport and Peer-to-Peer (P2P) systems built with web services, but the plumbing that enables these systems will never be sexy.

Part of the fundamental utility of web services is their language independence—we come back to this again and again in the book. We show how Java, Perl, C#, and Visual Basic code can be easily integrated using the web services architecture, and we describe the underlying principles of the web service technologies that transcend the particular programming language and toolkit you choose to use.

Audience for This Book

There's a shortage of good information on web services at all levels. Managers are being bombarded with marketing hyperbole and wild promises of efficiency, riches, and new markets. Programmers have a bewildering array of acronyms thrust into their lives and are expected to somehow choose the correct system to use. On top of this confusion, there's pressure to do something with web service immediately.

If you're a programmer, we show you the big picture of web services, and then zoom in to give you low-level knowledge of the underlying XML. This knowledge informs the detailed material on developing SOAP web services. We also provide detailed information on the additional technologies needed to implement enterprise-quality web services.

Managers can benefit from this book, too. We strip away the hype and present a realistic view of what is, what isn't, and what might be. Chapter 1 puts SOAP in the wider context of the web services architecture, and Chapter 9 looks ahead to the future to see what is coming and what is needed (these aren't always the same).

Structure of This Book

We've arranged the material in this book so that you can read it from start to finish, or jump around to hit just the topics you're interested in.

Chapter 1, places SOAP in the wider picture of web services, discussing Just-in-Time integration and the Web Service Technology Stack.

Chapter 2, explains what SOAP does and how it does it, with constant reference to the XML messages being shipped around. It covers the SOAP envelope, headers, body, faults, encodings, and transports.

Chapter 3, shows how to use SOAP toolkits in Perl, Visual Basic, Java, and C# to create an elementary web service.

Chapter 4, presents our first real-world web service. Registered users may add, delete, or browse articles in a database.

Chapter 5, introduces the Web Services Description Language (WSDL) at an XML and programmatic level, shows how WSDL makes it easier to write a web service client, and discusses complex message patterns.

Chapter 6, shows how to use the Universal Description, Discovery, and Integration (UDDI) project and the WS-Inspection standard to publish, discover, and call web services, and features best practices for using WSDL and UDDI together.

Chapter 7, builds a peer-to-peer (P2P) web services application for sharing source code in Perl and Java using SOAP, WSDL, and related technologies.

Chapter 8, describes the issues and approaches to security in web services, focusing on Microsoft Passport, XML Encryption, and Digital Signatures.

Chapter 9, explains the present shortcomings in web services technologies, describes some developing standardization efforts, and identifies the future battlegrounds for web services mindshare.

Appendix A, is a summary of the many varied standards for aspects of web services such as packaging, security, transactions, routing, and workflow, with pointers to online sources for more information on each standard.

Appendix B, is a gentle introduction to the bits of the XML Schema specification you'll need to know to make sense of WSDL and UDDI.

Appendix C, contains full source for the programs developed in this book.

Conventions

The following typographic conventions are used in this book:

Italic

Used for filenames, directories, email addresses, and URLs.

`Constant Width`

Used for XML and code examples. Also used for constants, variables, data structures, and XML elements.

`Constant Width Bold`

Used to indicate user input in examples and to highlight portions of examples that are commented upon in the text.

`Constant Width Italic`

Used to indicate replaceables in examples.

Comments and Questions

We have tested and verified all of the information in this book to the best of our ability, but you may find that features have changed, that typos have crept in, or that we have made a mistake. Please let us know about what you find, as well as your suggestions for future editions, by contacting:

O'Reilly & Associates, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
(800) 998-9938 (in the U.S. or Canada)
(707) 829-0515 (international/local)
(707) 829-0104 (fax)

You can also send us messages electronically. To be put on the mailing list or request a catalog, send email to:

info@oreilly.com

To ask technical questions or comment on the book, send email to:

bookquestions@oreilly.com

We have a web site for the book, where we'll list examples, errata, and any plans for future editions. You can access this page at:

http://www.oreilly.com/catalog/progwebsoap/

For more information about this book and others, see the O'Reilly web site:

http://www.oreilly.com/

Acknowledgments

The authors and editor would like to thank the technical reviewers, whose excellent and timely feedback greatly improved the book you read: Ethan Cerami, Tony Hong, Matt Long, Simon Fell, and Aron Roberts.

James

Thank you,

To Pavel and Doug, for their help.
To my editor, Nathan, for his persistent badgering.
To my wife, Jennifer, for her patience.
To my son, Joshua, for his joy.
And to my God, for his grace.

This book wouldn't exist without them.

Doug

I would like to thank my wonderful wife, Sheri Castle, and our amazing daughter, Lily, for their love and support. Nothing I do would be possible or meaningful without them.

Pavel

I wouldn't have been able to participate in this project without my family's patience and love. My son, Daniil, was the source of inspiration for my work, and my wife, Alena, provided constant support and encouragement. Thank you!

Many thanks to Tony Hong for his sound technical advice, productive discussions, and our collaboration on projects that gave me the required knowledge and experience.

I'd like to thank James Snell for inviting me to participate in writing this book, and for the help he gave me throughout the process.

Thanks to our wonderful technical editor, Nathan Torkington, who was a delight to work with and wonderfully persistent in his efforts to get this book done and make it great.

Finally, I am fortunate to be part of two communities, Perl and SOAP. I want to thank the many people that make up those communities for the enthusiastic support, feedback, and the fresh ideas that they've provided to me—they've helped to make SOAP::Lite and the other projects I've worked on what they are now.